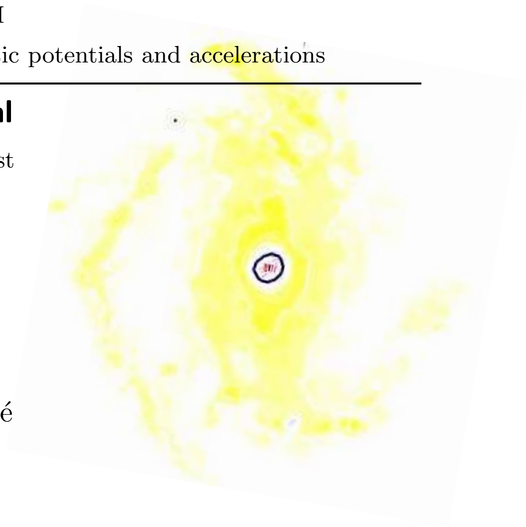
Brainstorm

An easy-to-use F90-program to compute galactic potentials and accelerations

User's manual

2005, September 1st



Jean-Marc Huré

1 Overview of BRAINSTORM capabilities

BRAINSTORM is an easy-to-use Fortran 90 program devoted to the study of *galaxy dynamics*. It reads and write *FITS format* files thanks to the CFITSIO numerical library. The *gravitational potential and field components* are then computed in the rotation plane of galaxies as well as oustide it from the *splitting method* (Huré 2005). In contrast with most methods, this one uses no kind of approximation or modification of Newton's law (like smoothing length or so). Matter is assumed to be *continuously* distributed over a planar, finite size rectangular surface. Singularities in the Poisson kernels are exactly treated in the homogeneous limit (see the appendix for a brief outline of the splitting method in Cartesian coordiantes). A bi-dimensional dynamical mesh ensures reliability, high accuracy and a not-so-long computing time (typical of integral based-techniques).

BRAINSTORM is part of the NOBONES project (by the same author) to be released soon.

2 An example

Figure 1 shows an optical, deprojected image of the galaxy NGC 4303, a FITS format image (561×561 pixels). This image is mapped into a square $[-1, +1]$ in both X- and Y-directions. The zone where the potential is requested is marked by a white square. It is $[-0.2, +0.2]$ in both directions. The selection can be a single point, a surface, a cube, a coplanar or an off plane region. The number of pixels inside this computational box is also user-defined. Here, it is a 64×64 box (the more the pixels, the longer the computing time).

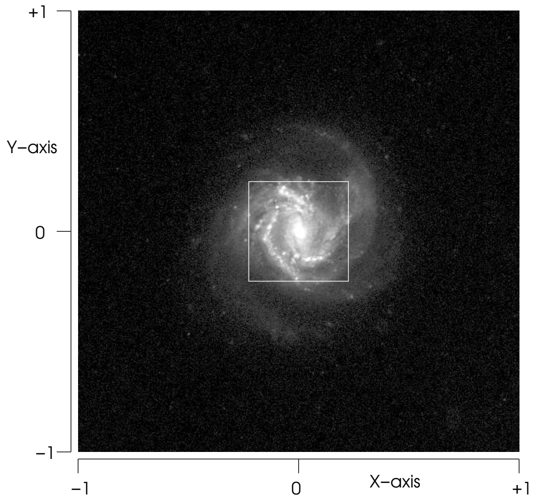


Fig. 1. Optical g-band image of the galaxy NGC4303 from *The Galaxy Catalog* by Z. Frei & J.E. Gunn (see `http://www.astro.princeton.edu/ frei/galaxy_catalog.html`. This FITS format image is a square image with 561 × 561 pixels. The small white square is the selected computational box.

Figure 2 displays the gravitational potential (log. scale) and the rotational energy (normal scale) in the plane of the galaxy NGC 4303 as output by Brainstorm. The computing time is of the order of one minute on a low cost 1.5 GHz personal computer (no optimization option, no paralellization). Brainstorm also computes the error on the computed potential as well as field components (not shown here).

3 Copyright

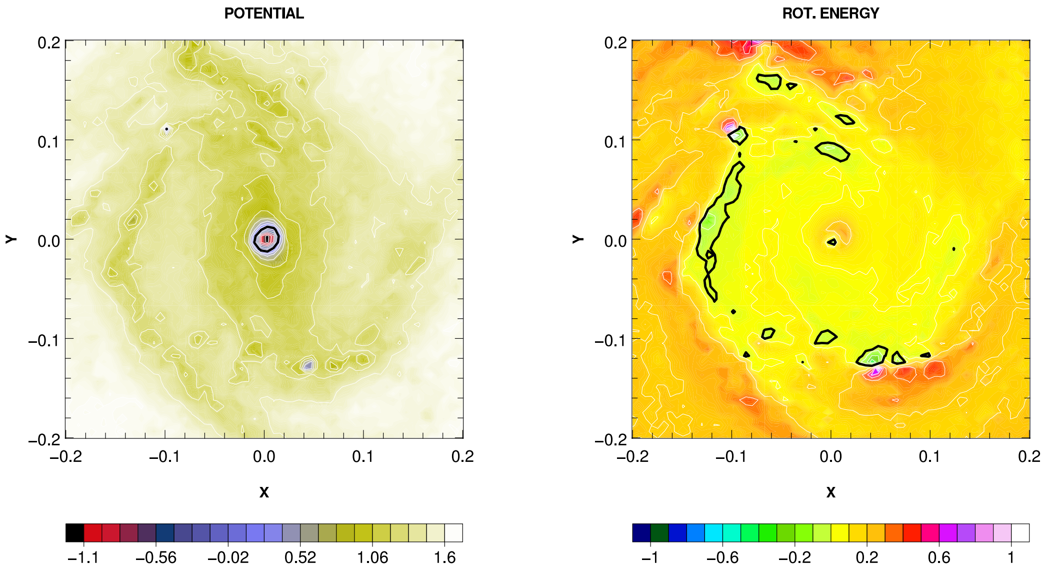


Fig. 2. *Left*: gravitational potential (normalized, log. scale) computed with Brainstorm. *Right*: steady specific kinetic energy (normalized, linear scale).

330, Boston, MA 02111-1307, USA.

4 Author address

Jean-Marc Huré,

Observatoire Aquitain des Sciences de l'Univers,
2, rue de l'Observatoire,
B.P. 89,
33270 Floirac,
France

and

Université Bordeaux 1,
351, cours de la Libération,
33405 Talence cedex,
France

email: `jean-marc.hure@obs.u-bordeaux1.fr`

5 Download and installation

Brainstorm can be downloaded at the following Internet address

```
http://www.obs.u-bordeaux1.fr/radio/pages_web_radio/web_hure/
```

Two files are required:

- the UNIX compressed archive file, nammed `brainstorm.tar.gz`. It includes the source code (several modules and a demo program written in Fortran 90), a FITS format file, a default batch file `data.brainstorm`, a `Makefile`, and this user manual in PDF format.

- the installation script, nammed `installbrainstorm`.

Once these two files have been downloaded, you can install the program. Make the installation script executable, like this:

```
chmod +x installbrainstorm
```

and run it with the command:

```
./installbrainstorm
```

This uncompresses the archive, creating a directory nammed `BRAINSTORM` and several files within it. In principle, you should need to modify none of the F90 routines and modules, except the demonstration program `brainstorm.f90`, the batch file `data.brainstorm` and the `Makefile`.

Reach this directory with the command:

```
cd BAINSTORM
```

6 Dependancy: the CFITSIO numerical library

The main program calls a few subroutines belonging to the CFITSIO numerical library which enables to read and write FITS format file. So, you need to install that library before compilation. It is free. You can find it at the Internet site:

```
http://heasarc.gsfc.nasa.gov/docs/software/
```

This provides a library nammed `libcfitsio.a` in your environment. In the following, we shall assume that this library is correctly installed in the directory `/usr/lib/`, as usual (otherwise, you must know the absolute path where the library is stored).

7 Compilation

Edit the `Makefile`. Enter the name of your Fortran 90 Compiler at the right place, for instance:

```
FCOMPL=f90
```

Possibly, change the absolute path where to find the CFITSIO numerical library, for instance:

```
CFITSIO=/usr/mylibs/
```

Compile and link the source code with the command:

```
make brainstorm
```

After the prompt, you can run the main demo program like this:

```
brainstorm.exe < data.brainstorm
```

where `data.brainstorm` is a batch file containing default parameters. If you wish to enter all the parameters one-by-one by hand, then just enter the command:

```
brainstorm.exe
```

...but this may be non very confortable for repetitive uses!

You can compile, link and run the source code with the command:

```
make rbrainstorm < data.brainstorm
```

8 Main intput parameters

Input parameters are for:

- the filenames (input/output),
- the object,
- the computation of Ψ and $\vec{g}$ (namely the accuracy of the double integrals),
- the computational box (size and resolution; see Fig. 3),

These input parameters are contained in the batch file `data.brainstorm`. This batch file must be of the following form (line numbers are added for convenience):

```
line 1            >>> FITS format file
line 2            n4303_pr.fits
line 3            >>> Mass/Luminosity conversion factor
line 4            1.0000E+00
line 5            >>> Distance (in kiloparsecs)
line 6            750.
line 7            >>> Pixel size (in deg.)
```

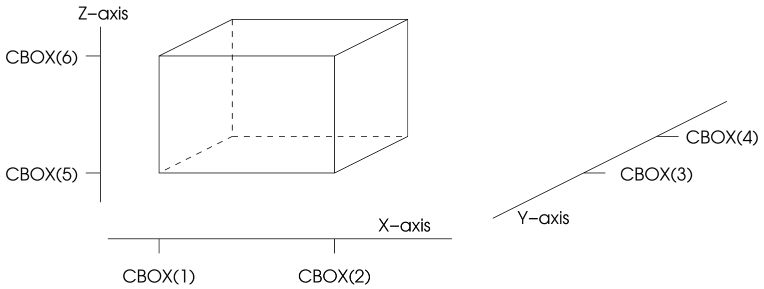


Fig. 3. The computational box (X, Y, Z).

```
line 8           7.4E-4
line 9           >>> Generic name for output files [auto,*]
line 10          auto
line 11          >>> Number of source point, x-axis
line 12          33
line 13          >>>> Number of source point, y-axis
line 14          37
line 15          >>> Computational box [auto,zoom,full]
line 16          zoom
line 17          >>> x-axis box; sampling NBOX and edges CBOX(1) CBOX(2)
line 18          132
line 19          -0.15E0
line 20          +0.11E0
line 21          >>> y-axis box; sampling LBOX and edges CBOX(3) CBOX(4)
line 22          134
line 23          -0.12E0
line 24          +0.23E0
line 25          >>> z-axis box; sampling MBOX and edges CBOX(5) CBOX(6)
line 26          3
line 27          +0.0E0
line 28          +0.3E0
```

All lines beginning with `>>>` are unread line (or used for comment only). Let us analyze this file line-by-line:

- `n4303_pr.fits` at line 2 is the name of the FITS format file containing the image. This file must has the suffix ".fits"

- `1.0000E+00` at line 4 is the mass/luminosity ratio (the flux or count must be converted into surface density)

- `750.` at line 6 is the remote distance from the galaxy expressed in kiloparsecs.

- `7.4E-4` at line 8 is the angular size of a image pixel
- with `auto` at line 10, output filenames are of the form `n4303_pr.*`. Any other filename works.
- `33` at line 12 is the number of source point used to perfom numerical integration in the x-direction; if you decrease this value, then the precision will fall down. The larger this value the larger the computing time, the more accurate the potential.
- `37` at line 14 is the number of source point used to perfom numerical integration in the y-direction; if you decrease this value, then the precision will fall down. The larger this value the larger the computing time, the more accurate the potential.
- with `auto` at line 16, the computational box automatically fits the genuine image, with the same distribution of pixels (lines 18 to 28 herebelow are then obsolete). Three other values are allowed here. With `fine`, the computational box automatically fits the genuine image but you can vary the number of pixels in each direction (lines 19, 20, 23 and 24 herebelow are then obsolete). With `zoom`; the computational box still lies in the image plane, but can be shorter/wider than the image (lines 26 to 28 herebelow are then obsolete); with `full`, you specify the full computational box at lines 18 to 28.
- `132` at line 18 is the number of point of the computational box in the X-direction, namely `NBOX`. It can be 1.
- `-0.15` at line 19 is the X-coordinate of the inner edge of the computational box, namely `CBOX(1)`.
- `+0.11` at line 20 is the X-coordinate of the outer edge of the computational box, namely `CBOX(2)`.
- `134` at line 22 is the number of point of the computational box in the Y-direction, namely `LBOX`. It can be 1.
- `-0.12` at line 23 is the Y-coordinate of the inner edge of the computational box, namely `CBOX(3)`.
- `+0.23` at line 24 is the Y-coordinate of the outer edge of the computational box, namely `CBOX(4)`.
- `1` at line 26 is the number of point of the computational box in the Z-direction, namely `MBOX`. It can be more than 1.
- `+0.0` at line 27 is the Z-coordinate of the bottom edge of the computational box, namely `CBOX(5)`.
- `+0.3` at line 28 is the Z-coordinate of the top edge of the computational box, namely `CBOX(6)`.

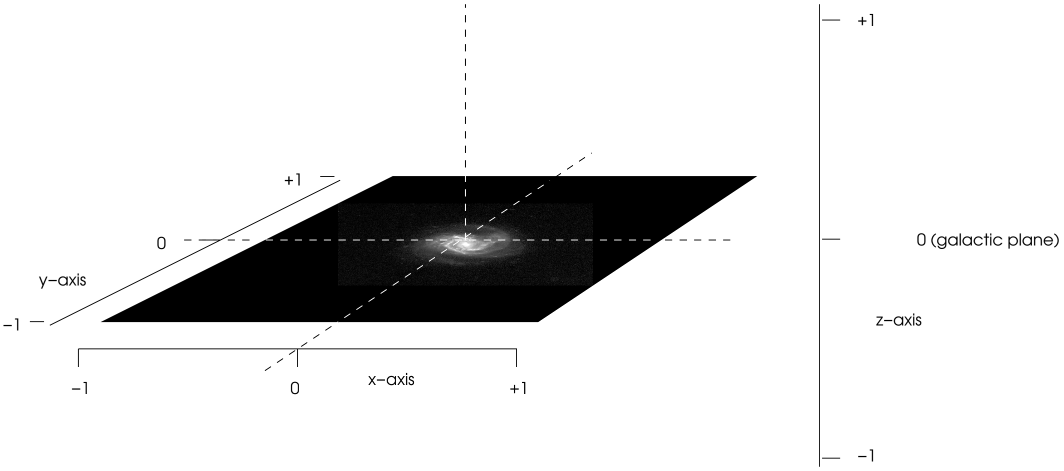


Fig. 4. Coordinate system (x, y, z) linked to the source. Here the image is square.

9 Calling sequence

Within the main program `brainstorm.f90`, the F90 calling sequence is the following:

```
!   inputs
XF=3.0
YF=-0.3
ZF=1.123
N=65
L=73
WHICH=1
!   calling sequence
CALL NOBONES2XYZ(XF,YF,ZF,N,L,WHICH,X,XERROR)
!   output
PSI=X(1)
GX=X(2)
GY=X(3)
GZ=X(4)
```

where:

- `XF`, `YF` and `ZF` denote the coordinates of the field point belonging to the computational box, i.e. X, Y and Z.
- `N` and `L` are the number of source points effectively used in the computation of the double integrals. `N` is for the X-axis, and `L` is for the Y-axis. The larger these numbers, the more accurate the results, the larger the computing time.

- `WHICH=1` orders the computation of the potential only. Set this varaibale to 2 for the field only, and to 12 for both.
- `X` is a vector: `X(1)` is the potential Ψ, `X(2)` is g_X, `X(3)` is g_Y and `X(4)` is g_Z.
- `XERROR` is a vector containing an estimate of the absolute error on `X`: `XERROR(1)` is the error on potential Ψ, `XERROR(2)` is g_X, `XERROR(3)` is g_Y and `XERROR(4)` is g_Z.

References

Huré J.-M., 2005, "Solutions of the axi-symmetric Poisson equation from elliptic integrals. I. Numerical splitting methods", *Astronomy and Astrophysics*, 434, 1

A Brief outline of the splitting method

The gravitational potential Ψ and acceleration $\vec{g} = -\nabla\Psi$ due to matter distributed over a surface $\mathcal{S}$ write in the integral forms

$$\Psi(\vec{r}) = -G \iint_{\mathcal{S}} \frac{\Sigma(\vec{r'})d^2r'}{|\vec{r'} - \vec{r}|}, \tag{1.1}$$

and

$$\vec{g}(\vec{r}) = G \iint_{\mathcal{S}} \frac{\Sigma(\vec{r'})(\vec{r'} - \vec{r})d^2r'}{|\vec{r'} - \vec{r}|^3}, \tag{1.2}$$

respectively, where Σ is the surface density, $\vec{r'}$ is the spherical vector refering to location of matter (i.e. the source), and $\vec{r}$ is for the field point (i.e. the point of space where the information are requested). It is worth noting that the bare inversion of the Poisson equation in two dimension (2D) would require the knowledge of the discontinuity of the vertical gravity field, namely $\partial^2_{zz}\Psi$. This quantity is generally not straigtforwardly available, that is why there is no way around the integral approach for 2D objects like those considered here. The difficulty with the integral formalism is the presence of singular integrands: everwhere on the source, $\vec{r'} = \vec{r}$ does occur in Eqs.(1.1) and (1.2), making the straight integration of these expressions not easy (if some precision is needed). Well outside the source, there is not such difficulty since the equality $\vec{r'} = \vec{r}$ never occur in the above expressions.

If the surface has finite size, is planar and of rectangular shape, then Cartesian coordinates are appropriate. The potential is then

$$\Psi(X,Y,Z) = -G \iint_{\mathcal{S}} \frac{\Sigma(x,y)dxdy}{\sqrt{(x-X)^2 + (y-Z)^2 + (z-Z)^2}}, \tag{1.3}$$

where (x,y,z) are associated to the source and (X,Y,Z) are for the field point. Unfortunately, this double integral generally does not lead to a closed form (i.e. a formula involving a finite number of analytical terms). A remarkable exception however is the *homogeneous rectangular sheet.* Actually, with $\Sigma(x,y) = cst$, the potential is indeed fully analytical (we shall not give the formula here). We can advantageously use this issue to numerically compute with great accuracy the potential due to any planar distribution *inside the distribution.* How ? For any field point $(X,Y,Z=z)$ located onto the surface, we can always form the difference surface density

$$\delta\Sigma(x,y;X,Y) = \Sigma(x,y) - \Sigma(X,Y). \tag{1.4}$$

Then Ψ writes

$$\Psi(X,Y,Z) = -G \iint_{\mathcal{S}} \frac{[\Sigma(X,Y) + \delta\Sigma]\,dxdy}{\sqrt{(x-X)^2 + (y-Z)^2 + (z-Z)^2}}, \tag{1.5}$$

In other words, the potential is the sum of two contributions: i) the potential $\Psi_{\rm us}$ due to a uniform sheet with surface density $\Sigma(X,Y)$, namely the value

$$\Psi_{\rm us}(X,Y,Z) = -G\Sigma(X,Y) \iint_{\mathcal{S}} \frac{dxdy}{\sqrt{(x-X)^2+(y-Z)^2+(z-Z)^2}}, \tag{1.6}$$

and ii), the residual potential $\delta\Psi$ due to the distribution $\delta\Sigma$ (the difference bewteen the true surface density and the local value), namely

$$\delta\Psi(X,Y,Z) = -G \iint_{\mathcal{S}} \frac{\delta\Sigma dxdy}{\sqrt{(x-X)^2+(y-Z)^2+(z-Z)^2}}, \tag{1.7}$$

Since $\delta\Sigma = 0$ at the field point, the integral in $\delta\Psi$ has a regular integrand, contrary to the integral in Eq.(1.1). This means that Ψ can be computed numerically with great precision (depending on the numerical quadrature). Similar considerations hold for the gravitational field $\vec{g}$.

BRAINSTORM computes Ψ (and $\vec{g}$) on any rectangular *inhomogeneous sheet* using this efficient technique. The splitting method in cylindrical plar coordinates is described in details in Huré (2005).